Simplex Optimization

V. Zitko

Biological Station,
St. Andrews, N. B., E0G 2X0

September 1984

Canadian Technical Report of Fisheries and Aquatic Sciences No. 1308

35-1994/

Canadian Technical Report of Fisheries and Aquatic Sciences

These reports contain scientific and technical information that represents an important contribution to existing knowledge but which for some reason may not be appropriate for primary scientific (i.e. *Journal*) publication. Technical Reports are directed primarily towards a worldwide audience and have an international distribution. No restriction is placed on subject matter and the series reflects the broad interests and policies of the Department of Fisheries and Oceans, namely, fisheries management, technology and development, ocean sciences, and aquatic environments relevant to Canada.

Technical Reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report will be abstracted in *Aquatic Sciences and Fisheries Abstracts* and will be indexed annually in the Department's index to scientific and technical publications.

Numbers 1-456 in this series were issued as Technical Reports of the Fisheries Research Board of Canada. Numbers 457-714 were issued as Department of the Environment, Fisheries and Marine Service, Research and Development Directorate Technical Reports. Numbers 715-924 were issued as Department of Fisheries and the Environment, Fisheries and Marine Service Technical Reports. The current series name was changed with report number 925.

Details on the availability of Technical Reports in hard copy may be obtained from the issuing establishment indicated on the front cover.

Rapport technique canadien des sciences halieutiques et aquatiques

Ces rapports contiennent des renseignements scientifiques et techniques qui constituent une contribution importante aux connaissances actuelles mais qui, pour une raison ou pour une autre, ne semblent pas appropriés pour la publication dans un journal scientifique. Il n'y a aucune restriction quant au sujet, de fait, la série reflète la vaste gamme des intérêts et des politiques du Ministère des Pêches et des Océans, notamment gestion des pêches, techniques et développement, sciences océaniques et environnements aquatiques, au Canada.

Les Rapports techniques peuvent être considérés comme des publications complètes. Le titre exact paraîtra au haut du résumé de chaque rapport, qui sera publié dans la revue *Aquatic Sciences and Fisheries Abstracts* et qui figurera dans l'index annuel des publications scientifiques et techniques du Ministère.

Les numéros 1-456 de cette série ont été publiés à titre de Rapports techniques de l'Office des recherches sur les pêcheries du Canada. Les numéros 457-714, à titre de Rapports techniques de la Direction générale de la recherche et du développement, Service des pêches et de la mer, ministère de l'Environnement. Les numéros 715-924 ont été publiés à titre de Rapports techniques du Service des pêches et de la mer, Ministère des Pêches et de l'Environnement. Le nom de la série a été modifié à partir du numéro 925.

La page couverture porte le nom de l'établissement auteur où l'on peut se procurer les rapports sous couverture cartonnée.

Canadian Technical Report of
Fisheries and Aquatic Sciences 1308

September 1984

SIMPLEX OPTIMIZATION

by

V. Zitko
Fisheries and Environmental Sciences
Fisheries Research Branch
Department of Fisheries and Oceans
Biological Station
St. Andrews, N.B. EOG 2XO Canada

This is the one hundred and seventy-fifth Technical Report from
the Biological Station, St. Andrews, N. B.

©Minister of Supply and Services Canada 1984

Cat. No. FS 97-6/1308E ISSN 0705-6457

Correct citation for this publication:

Zitko, V. 1984. Simplex optimization. Can. Tech. Rep. Fish. Aquat. Sci. 1308: iii + 18 p.

ABSTRACT

Zitko, V. 1984. Simplex optimization. Can. Tech. Rep. Fish. Aquat. Sci. 1308: iii + 18 p.

The simplex optimization algorithm is reviewed and a program written in HP-BASIC is presented. The program includes several recent modifications of the simplex algorithm and options for data input, initial simplex selection, summary report, and study of the optimized function.

RÉSUMÉ

Zitko, V. 1984. Simplex optimiztion. Can. Tech. Rep. Fish. Aquat. Sci. 1308: iii + 18 p.

Ce rapport contient une revue de l'algorithme simplex d'optimisation ainsi qu'un programme machine écrit en HP-BASIC. Le programme inclut plusieurs modifications récentes apportées à l'algorithme simplex, de même que diverses options pour l'entrée des donneés. Il comprend en outre une sélection simplex initiale, un résumé de rapport et une étude de la fonction optimisée.

INTRODUCTION

The objective of this report is to provide a versatile simplex program for general laboratory applications ranging from curve fitting to tuning of chemical analytical instruments.

Simplex optimization is a direct search method to find an extreme response. It is "direct" since it does not require the determination of slope(s) and "search" because it does not need a description of the response surface. The extreme response is most frequently a minimum response (for example, minimum deviation in fitting functions to data). It may be maximum response when dealing with the yield of a chemical reaction, instrument signal or peak separation in chromatography.

The examples above indicate the variety of laboratory applications such as chromatographic separation improvement by peak resolution (Cela and Perez-Bustamante 1983) or by actual separation optimization (Berridge 1984), reaction yield improvement (Dean et al. 1975), analytical methods development (Morgan and Deming 1974; Backman and Shanbag 1984), instrumental parameters optimization (Routh et al. 1977; Leggett 1983; Burgess and Hayumbu 1984), drug design (Darvas et al. 1978), and pattern recognition (Ritter et al. 1975; Kaberline and Wilkins 1978).

Computer programs for simplex optimization are not readily available. A FORTRAN listing based on the original algorithm was published (O'Neill 1971). A modified version including I/O procedures was developed by Olsson (1974). The Tektronix statistics package (Tektronix 1975) contains a version of the original algorithm in Tektronix BASIC.

The newer simplex algorithms are available in an ALGOL-like notation from Ryan et al. (1980) and commercial programs are obtainable from van der Wiel et al. (1983). Simplex algorithms are also included as parts of pharmacokinetic programs (Yamaoka et al. 1981; Zitko 1982).

Surprisingly, neither SPSS (Nie et al. 1975) nor IMSL (IMSL 1982) contains simplex-based optimization programs.

SIMPLEX ALGORITHM

Simplex optimization was presented to analytical chemists by Deming and Morgan (1973) and to fisheries biologists by Schnute (1982). A concise description by Shavers et al. (1979) is available and almost all the papers quoted contain a brief summary of simplex search rules.

Simplex is an n-dimensional object with (n+1) vertices; a triangle in two, a tetrahedron in three, dimensions. The vertices are sets of values of parameters (n parameters per vertex) at which the response is determined. The best (B), worst (W) and next-to-worst (N) vertices are located. The mean of all vertices except W is the centroid (C, Fig.1). Points NC, C, PC, R, and E are negative contraction, centroid, positive contraction, reflection, and expansion, respectively. The mean of all vertices (not shown) is the center of gravity.

As can be seen from Fig. 1, all the points of potential simplex movement (NC, PC, R, and E) lie on the straight line through W and C, given by Eq. 1:

(Eq. 1) $$\text{Point} = C + F*(C-W),$$

where Point=NC, PC, R, or E and F=-0.5, 0.5, 1 or 2, respectively.

Consider for example a two-dimensional simplex of the function z = 2 - x**2 - y**2:

x	1	3	2
y	1	1	3
z	0	-8	-11

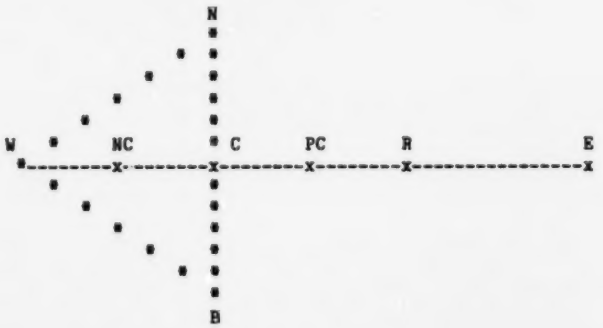

Fig. 1. A two-dimensional simplex. Responses in vertices are: worst (W), next-to-worst (N), and best (B). NC, C, PC, R, and E indicate negative contraction, centroid, positive contraction, reflection, and expansion, respectively. For simplex rules, see text and Fig. 2.

The best response (O) is at x=y=1; the worst (-11) is at x=2, y=3. The coordinates of C are x=2 and y=1 [(1+3)/2;(1+1)/2] and those of R are x=2 and y=-1 [2+(2-2);1+(1-3)].

The simplex search rules have changed somewhat over the years, mainly as a result of modifications suggested by O'Neill (1971), Brookes et al. (1981), and Aberg and Gustavsson (1982). The algorithm outlined in Fig. 2 is that of Aberg and Gustavsson (1982), except for the "RSP.BT." test. This has been added to stop the algorithm when there is no further improvement in response at the vertices. The addition of this test improved the performance of the algorithm in optimizations described later in this report.

SIMPLEX MODIFICATIONS

Super Modified Simplex (SMS, Routh et al. 1977) locates the worst vertex (W), centroid (C), and reflection (R). It determines the responses at C and R (that at W is available), and fits a second degree polynomial to these three values. A new vertex is then located at the minimum (maximum) of this polynomial or close to a preselected boundary along the W-C-R line. If the response at the latter is not better than that at R, the response at R is retained. In many situations SMS appears to be faster than the usual simplex algorithm.

Additional improvement in speed is achieved by modifications in calculating the centroid (Brissey et al. 1979). According to these authors, center of gravity rather than the centroid is calculated for the initial simplex. It is then updated as required by adding the fraction of the difference between the new and the old vertex (Eq. 2):

(Eq. 2) [Center gravity]new=[Center gravity]old + ([Vertex]new-[Vertex]old)/n

n = number of vertices.

Centroid need not be calculated since it can be expressed in terms of center of gravity and the worst vertex (Eq. 3):

(Eq. 3) Centroid=(n*[Center gravity]-[Worst vertex])/(n-1)

The new vertex is located on the line determined by the centroid and the worst vertex (Eq. 4):

(Eq. 4) [Vertex]new=[Worst vertex] + F*(Centroid-[Worst vertex])

F = 0.5, 1.5, 2, or 3 for negative contraction, positive contraction, reflection, or expansion, respectively, in the standard simplex, or is determined from the quadratic polynomial in SMS. Eq. 3 can be used instead of explicit calculation of centroid in Eq. 4.

Another modification includes the weighting of vertices during the calculation of the centroid (Keefer 1973; Ryan et al. 1980). This technique may lead to faster convergence by providing a more direct path to the extreme sought. According to some authors (see, for example, Aberg and Gustavsson 1982) this modification does not outperform the conventional simplex, particularly with noisy data.

PROGRAM DESCRIPTION

The program SX (see listing in Appendix 1) incorporates the modifications expressed by Eq. 2-4. The simplex algorithm is called as a subroutine from a master program. The latter establishes operational parameters (type of extreme, number of simplex evaluations, convergence criterium, and frequency of printing intermediate results, etc.). The default values are minimum, 120, 1E-4, and 20, respectively. Data input may be from file (for format see below), manually, or, in the case of finding the minimum of a function, data are not needed. The master program also selects the initial simplexes (several, default 10) to ensure that the simplex convergence is consistent. Initial simplexes may be written to a file together with the final best value. This may be useful in studying the properties of the optimized function.

An example of the interactive input is presented in Appendix 2. For stereotype routine applications it is of advantage to change the INPUT statements into REMarks and to establish the desired values of the variables in the statement following the INPUT statement. Whenever possible, the program should be operated in compiled BASIC.

Appendix 2 also contains an example of the output for function number 4, described later in this report. The output gives the number of attempts needed to establish the initial simplex within the specified range of variables and the actual initial simplex. Following this, the letter codes record operations performed on the simplexes. E, R, PC, and NC are as defined, Prn and Nrn indicate positive (negative) contraction followed by reflection from the next-to-worst vertex. Interim results (simplex and responses) are printed next, followed by the final result ("converged" or "not converged"). The program finishes by printing a summary table.

DATA INPUT

Standard data file for curve fitting has the format below:

```
1234567890123456      <--- column numbers
4                     <--- number of points
0              5      <--- independent and dependent
                          variable
1             10           (starting in columns 1
2             19           and 15, respectively)
3             32
```

The following format is needed to present an initial simplex from a file:

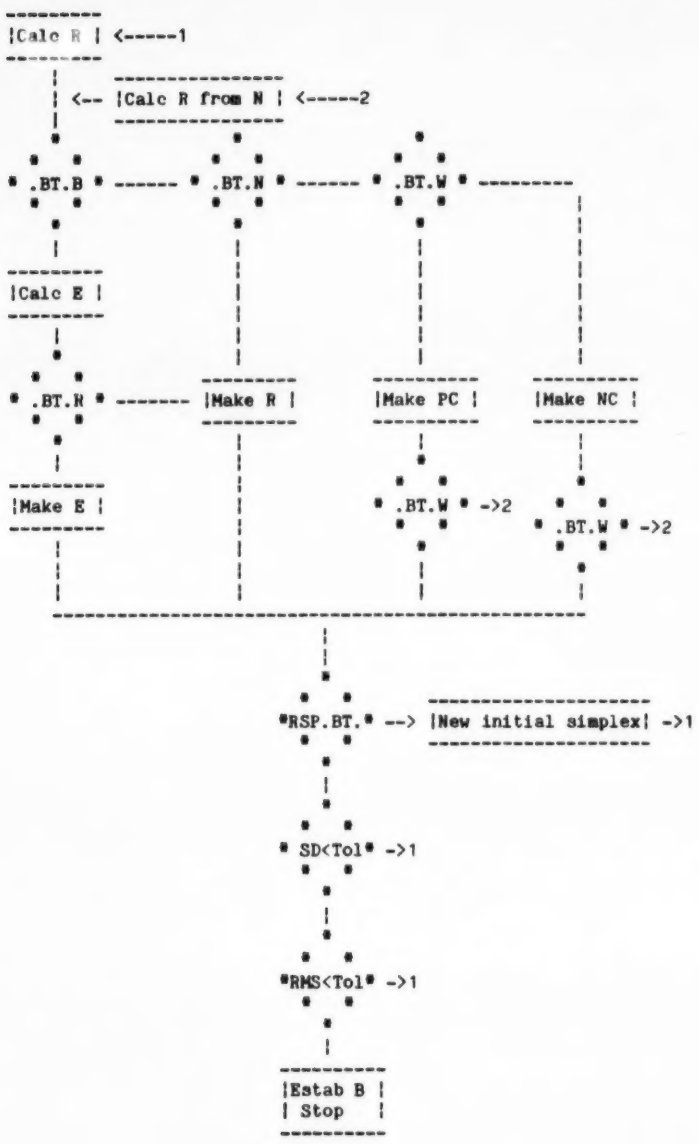

Fig. 2. Simplex search rules (after Aberg and Gustavsson 1982).

B,N,W = best, next-to-worst, worst response, respectively;
R,E,PC,NC = reflection, expansion, positive and negative contraction, respectively;
.BT. = better than; yes: down, no: right;
RSP = mean of responses at all vertices; in this case .BT. means better than previous;
SD = standard deviation of all responses;
Tol = tolerance, typically 0.0001;
RMS = root mean square deviation from response at the center of gravity;
Calc = calculate coordinates;
Make = substitute vertex of worst response by new vertex.

```
1234567890123456789
PAR                    <--- Identification as initial
                            simplex file
3                      <--- Number of parameters
1    2     1    3     <--- Parameters in columns
3    3     3    3
5    4     6    3
```

The initial simplex may also be located randomly within a certain range of parameter values. Aberg and Gustavsson (1982) used this technique when checking the performance of their simplex algorithm. This concept was expanded to N individually scaled parameters as follows:

The program selects N random numbers and scales them to the specified ranges. These become the coordinates of the center of gravity of the initial simplex.

Coordinates of the first vertex are:

$$X1l = X1g - D1/2$$
$$X2l = X2g - D2/3$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$Xnl = Xng - Dn/(n+1)$$

The other vertices are then calculated. If any of the vertices is outside the range originally specified, a new center of gravity is selected.

The initial simplex is located with its side parallel to the plane defined by the first (N-1) parameters. Within this plane, the (N-2)-dimensional edge lies parallel to the plane defined by the first (N-2) parameters, etc.

The initial 4-dimensional simplex is then (vertices arranged in columns; Xij is the i-th coordinate of j-th vertex; Di is the initial step on the i-th coordinate):

X1l	X1l+D1	(2X1l+D1)/2	(2X1l+D1)/2	(2X1l+D1)/2
X2l	X2l	X2l+D2	(3X2l+D2)/3	(3X2l+D2)/3
X3l	X3l	X3l	X3l+D3	(4X3l+D3)/4
X4l	X4l	X4l	X4l	X4l+D4

The modification of the above to n-dimensions can be readily accomplished.

SIMPLEX ALGORITHM

The program is kept streamlined by coding the simplex operations in the form of "multiline functions". These closely resemble FORTRAN subroutines and facilitate modifications and translations.

Programming input is required in specifying the "optimized" function with which the algorithm is going to deal. The multiline function for this task is FNV. One of three possibilities is normally encountered:

1. Function minimization

The function is coded in FNV to return value S as a function of parameters in the array T[*]. For example, to find the minimum of the function:

$$Z = X^{**}2 + Y^{**}2$$

the definition of FNV is in Fig. 3.

2. Interactive evaluation

This may be used to manually optimize instrumental response, chemical reaction, analytical method, etc.

Input of response at a presented setting of parameters (T[*]) is required. No input (just return) displays the status of the simplex and its response (Fig. 4).

This modification deals with the input buffer [BUF(0)] and in this sense is specific to the HP 3000 BASIC. Since it involves just a convenience, it is readily modifiable for computers on which the input buffer may not be accessible in this way.

3. Curve fitting

The program expects values of the independent variable in the array X1[*] and those of the dependent variable in the array V1[*]. The fitted function must be supplied in FNV (Fig. 5). The program then minimizes the sum of squared deviations between V1[*] and the values calculated from the fitted function at given values of the independent variable. For example, if the fitted function is

$$A = A(inf)*(1-exp(-kt)),$$

the fitted parameters A(inf) and k are stored as T[1] and T[2], respectively. The values of A and t are in the arrays V1[*] and X1[*] (see function 2 in the following section).

TESTING PROGRAM'S PERFORMANCE

The performance was tested by finding extremes of functions and by fitting one exponential function. The test functions were those used by Aberg and Gustavsson (1982):

1. Two-dimensional paraboloid $Z = X^{**}2 + Y^{**}2$

   Region: $-1000 <= X <= 1000$; $-1000 <= Y <= 1000$
   Minimum: X=0; Y=0; Z=0

2. Least-squares response function with parameters from Deming and Morgan (1973):

   $$Z = \text{Sum } [A - X*(1 - \exp(-Yt))]^{**}2$$

   Region $0 <= X <= 2$; $0 <= Y <= 2$
   Minimum: X=0.404; Y=0.170; Z=0.0036

Data for the least-squares response function (Deming and Morgan 1973):

t	A
1.5	0.110
1.5	0.109
3.0	0.169
3.0	0.172
4.5	0.210
4.5	0.210
6.0	0.251
6.0	0.255
9.0	0.331
9.0	0.325
12.0	0.326
12.0	0.330
15.0	0.362
15.0	0.383
18.0	0.381
18.0	0.372
24.0	0.422
24.0	0.411

```
4200 DEF FNV(P,S,G,T[*],V1[*],X1[*],V[*],X[*,*])
4210   REM Calculate function value
4220 REM ************************************************************
4230 REM                    EVALUATED FUNCTION COMES IN HERE      *
4240 REM Must return value in S
4250 S=T[1]**2+T[2]**2
4260 REM                                                          *
4270 REM ************************************************************
4280   RETURN S
4290 FNEND
```

Fig. 3. Function minimization.

```
4200 DEF FNV(P,S,G,T[*],V1[*],X1[*],V[*],X[*,*])
4210   REM Calculate function value
4220   INTEGER I
4230 REM ************************************************************
4240 REM                    EVALUATED FUNCTION COMES IN HERE      *
4250 REM Must return value in S
4260 FOR I=1 TO G
4270 PRINT T[I]
4280 NEXT I
4290 INPUT "The response is (CR displays status) ":
4300 X=BUF(0)
4310 IF X<>4 THEN 1885
4320 MAT PRINT X
4330 PRINT "--------"
4340 MAT PRINT V
4350 GOTO 1840
4360 REM                                                          *
4370 REM ************************************************************
4380 INPUT :S:
4390   RETURN S
4395 FNEND
```

Fig. 4. Interactive simplex evaluation.

```
4200 DEF FNV(P,S,G,T[*],V1[*],X1[*],V[*],X[*,*])
4210   REM Calculate function value
4220   INTEGER I
4230   S=0
4240 REM ************************************************************
4250 REM                    EVALUATED FUNCTION COMES IN HERE      *
4260 REM Must return value in S
4270 FOR I=1 TO P
4280 S=S+(V1[I]-T[1]*(1-EXP(-T[2]*X1[I])))**2
4290 NEXT I
4300 REM                                                          *
4310 REM ************************************************************
4320   RETURN S
4330 FNEND
```

Fig. 5. Curve fitting.

3. Rosenbrock's function Z = 100*(Y-X)**2 + (1-X)**2

   Region -25<=X<=25; -25<=Y<=25
   Minimum: X=1; Y=1; Z=0

4. Modified Fletcher and Powell's helical valley

```
Z = 100*[X - 10*Q]**2 +
    [(X**2+Y**2)**.5 -1 ]**2 +Y**2
```

where

```
Q=[ATN(Y/X)]/2*Pi for X>0
Q=[Pi + ATN(Y/X)]/2*Pi for X<0
```

Region -100<=X<=100; -100<=Y<=100
Minimum: X=0.691; Y=0.319; Z=0.159

In addition, the following function was used:

5. Z = 2*X**3 - 3*X**2 - 2*Y**3 + 3*Y**2

   Region -2<=X<=2; -2<=Y<=2
   Minimum X=1; Y=0; Z=-1
   Maximum X=0; Y=1; Z=1

The algorithm performed well on all the test functions (Table 1). The number of evaluations needed for convergence almost doubled with decreasing initial simplex size (ISS). This agrees with the results of Aberg and Gustavsson (1982). The convergence was good, even for function number 5. The extremes of this function are "local". Since the optimization was performed without constraints on the variables, the nonconverging simplexes were those that failed "to home in" on the local extremes.

Table 1. Results of tests on functions specified in text.

ISS	Paraboloid (1)			Least squares (2)			Rosenbrock (3)		
	Conv.*	Evaluation		Conv.*	Evaluation		Conv.*	Evaluation	
		Mean	SD		Mean	SD		Mean	SD
0.2	0/0/10	36.5	3.3	0/0/10	17.8	2.9	0/0/10	32.2	3.1
0.1	0/0/10	34.4	3.1	0/0/10	19.5	9.4	0/0/10	31.1	3.8
0.05	0/0/10	40.9	9.0	0/0/10	25.7	8.8	0/0/10	32.5	4.6
0.02	0/0/10	37.8	5.4	0/0/10	24.9	9.3	0/0/10	35.4	5.1
0.01	0/0/10	38.6	6.1	0/0/10	24.9	10.6	0/0/10	36.6	7.8
0.005	0/0/10	48.6	3.1	0/1/9	38.8	11.6	0/0/10	38.2	9.9
0.001	0/0/10	46.7	3.7	0/1/9	35.7	11.6	0/0/10	47.5	6.8
0.0005	0/0/10	54.8	7.1	0/2/9	39.2	12.0	0/0/10	54.0	5.8

ISS	Helical valley(4)			Minimum (5)			Maximum (5)		
	Conv.*	Evaluation		Conv.*	Evaluation		Conv.*	Evaluation	
		Mean	SD		Mean	SD		Mean	SD
0.2	0/0/10	50.0	6.0	6/0/9	19.3	1.8	5/0/10	18.0	2.2
0.1	0/0/10	54.7	8.2	10/0/9	18.2	2.4	8/0/9	17.0	2.4
0.05	0/0/10	45.5	3.7	7/0/9	19.4	3.8	0/0/10	19.6	2.7
0.02	0/0/10	57.3	15.2	3/0/10	19.2	3.3	9/0/9	21.8	3.0
0.01	1/0/10	57.2	11.6	8/0/9	23.7	3.9	6/0/6	24.9	9.1
0.005	0/1/9	63.3	18.5	12/0/8	25.4	4.7	9/0/9	27.0	7.5
0.001	0/0/10	64.2	15.5	13/1/8	31.1	7.2	5/0/10	33.6	4.5
0.0005	2/1/9	64.3	7.5	23/0/7	34.3	8.2	3/0/10	37.8	6.9

ISS = Initial simplex size as a fraction of range
Conv.= Number of: Not converged and detected/Not converged and not detected/Converged

ACKNOWLEDGMENTS

I thank Mrs. J. Hurley for formatting the manuscript, Mrs. R. Garnett for editorial assistance, Mrs. M. M. Irwin for proofreading, Mr. S. Bellis for explaining HP-manuals, the staff of the computer center for reliable maintenance of the system, and Dr. K. Haya and Mr. L. E. Burridge for comments.

REFERENCES

Aberg, E. R., and A. G. T. Gustavsson. 1982. Design and evaluation of modified simplex methods. Anal. Chim. Acta 144: 39-53.

Backman, L., and V. P. Shanbag. 1984. Simplex optimization in biochemistry: Application of the method in two-phase partition. Anal. Biochem. 138: 372-379.

Berridge, J. C. 1984. Automated multiparameter optimisation of high-performance liquid chromatographic separations using the sequential simplex procedure. Analyst 109: 291-293.

Brissey, G. F., R. B. Spencer, and C. L. Wilkins. 1979. High-speed algorithm for simplex optimization calculations. Anal. Chem. 51: 2295-2297.

Brookes, A. E., J. J. Leary, and D. W. Golightly. 1981. Modifications of the sequential simplex optimization method. Anal. Chem. 53: 720-721.

Burgess, D. D., and P. Hayumbu. 1984. Simplex optimization by advance prediction for single-element instrumental neutron activation analysis. Anal. Chem. 56: 1440-1443.

Cela, R., and J. A. Perez-Bustamante. 1983. Resolution of overlapping peaks in HPLC by means of the simplex algorithm. CAL 2: 137-144.

Darvas, F., L. Kovacs, and A. Eory. 1978. Computer optimization by the sequential simplex method in designing drug analogs. In: Quantitative Structure-Activity Analysis, R. Franke and P. Oehme, Editors, Abhandlungeg der Akademie der Wissenschaften der DDR, Nr.2N, 311-315.

Dean, W. K., K. J. Heald, and S. N. Deming. 1975. Simplex optimization of reaction yields. Science 189: 805-807.

Deming S. N., and S. L. Morgan. 1973. Simplex optimization of variables in analytical chemistry. Anal. Chem. 45: 278A-283A.

IMSL. 1982. IMSL Library Reference Manual. IMSL, Inc. Houston, Texas.

Kaberline, S. L., and C. L. Wilkins. 1978. Evaluation of the super-modified simplex for use in chemical pattern recognition. Anal. Chim. Acta 103: 417-428.

Keefer, D. L. 1973. Simpat: self-bounding direct search method for optimization. Ind. Eng. Chem. Process Des. Develop. 12: 92-99.

Leggett, D. J. 1983. Instrumental simplex optimization. J. Chem. Ed. 60: 707-710.

Morgan, S. L., and S. N. Deming. 1974. Simplex optimization of analytical chemical methods. Anal. Chem. 46: 1170-1181.

Nie, N. H., C. H. Hull, J. J. Jenkins, K. Steibrenner, and D. H. Bent. 1975. SPSS Statistical Package for the Social Sciences. Second Edition. McGraw-Hill Book Company, New York.

O'Neill, R. 1971. Algorithm AS47. Function minimization using a simplex procedure. Appl. Stat. 20: 338-345.

Olsson, D. M. 1974. A sequential simplex program for solving minimization problems. J. Quality Technol. 6: 53-57.

Ritter, G. L., S. R. Lowry, C. L. Wilkins, and T. L. Isenhour. 1975. Simplex pattern recognition. Anal. Chem. 47: 1951-1956.

Routh, M. W., P. A. Schwartz, and M. B. Denton. 1977. Performance of the Super Modified Simplex. Anal. Chem. 49: 1422-1428.

Ryan, P. B., R. L Barr, and H. D. Todd. 1980. Simplex techniques for nonlinear optimization. Anal. Chem. 52: 1460-1467.

Schnute, J. 1982. A manual of easy nonlinear parameter estimation in fishery research with interactive microcomputer programs. Can. Tech. Rep. Fish. Aquat. Sci. 1140: 115 p.

Shavers, C. L., M. L. Parsons, and S. N. Deming. 1979. Simplex optimization of chemical systems. J. Chem. Ed. 56: 307-309.

Tektronix Inc. 1975. PLOT 50 Statistics Vol.4. 3.1-3.9.

van der Wiel, P. F. A., R. Maassen, and G. Kateman. 1983. The symmetry-controlled simplex optimization procedure. Anal. Chim. Acta 153: 83-92.

Yamaoka, K., Y. Tanigawara, T. Nakagawa, and T. Uno. 1981. A pharmacokinetic analysis program (MULTI) for microcomputer. J. Pharm. Dyn. 4: 879-885.

Zitko, V. 1982. Two-compartment model of uptake and excretion of chemicals. Can. Tech. Rep. Fish. Aquat. Sci. 1076: 27 p.

Appendix 1 -- Listing of Program SX.

```
 10 FILES *,*
 20 X=RND(0)
 30 DIM R1[20],R2[20],R3[20],R4[20],X0[20,21]
 40 DIM X[20,21],V[21],T[20],T1[20],W1[20],C[20],G2[20]
 50 DIM D[5],D1[4]
 60 REM X[]-parameters,V[]-SUSQ values, T[],T1[]-temporary
 70 REM W1[]-worst, C[]-centroid,G2[]-center of gravity
 80 REM D[]- mean response,D1[]-mean response difference,"stacks"
 90 DIM X1[50],V1[50]
100 REM Fitted function indep variable values X1[],
110 REM                  dep variable values V1[]
120 DIM N$[8],N1$[8],N2$[8],N3$[8],B$[80],B1$[20]
130 REM N$,N1$,N2$,N3$-file names; B$,B1$- ASCII buffers
140 DIM S9[20,22]
150 REM S9[] is summary table
160 INTEGER C,C0,C2,C3,C5,I,I0,I5,I9,J,J5,K,G,G1,P,Q1,Q2,Q3,Q9,Z1
170 REAL S,Z,C1,W,N,B
180 LONG D
190 C0=C2=C4=0
200 C3=1
210 MAT D=ZER
220 MAT D1=ZER
230 REM INPUT "Minimum (1), maximum (2) ",C5
240 C5=1
250 INPUT "Desired function in FNV? (Y/N). IF NOT, c-Y or BRK ",Q$
260 REM INPUT "Number of evaluations ",I0
270 I0=120
280 REM INPUT "Convergence criterium (i.e. 1E-4) ",C1
290 C1=.0001
300 REM INPUT "Print results every n-th evaluation ",I7
310 I7=20
320 INPUT "Data from file (1), manually (2), not needed (3) ",Q1
330 IF Q1=1 THEN DO
340   INPUT "Input file ",N1$
350   ASSIGN N1$,1,Q9
360   GOSUB 4420
370   REDIM V1[P],X1[P]
380 DOEND
390 IF Q1=2 THEN DO
400   INPUT "Number of points ",P
410   REDIM X1[P],V1[P]
420   FOR I=1 TO P
430     INPUT "X ",X1[I]
440     INPUT "Y ",V1[I]
450   NEXT I
460 DOEND
470 IF Q1=3 THEN DO
480   P=0
490   MAT V1=ZER
500   MAT X1=ZER
510 DOEND
520 INPUT "Initial simplex from file(1),manually (2),computer(3) ",Q1
530 IF Q1=1 THEN DO
540   INPUT "Initial parameters file ",N2$
550   ASSIGN N2$,1,Q9
560   GOSUB 4420
570 DOEND
```

```
580 IF Q1<>1 THEN DO
590   INPUT "Number of parameters ",G
600 DOEND
610 G1=G+1
620 REDIM R1[G],R2[G],R3[G],R4[G],X0[G,G1]
630 REDIM X[G,G1],V[G1],T[G],T1[G],W1[G],C[G],G2[G]
640 IF Q1=2 THEN DO
650   FOR I=1 TO G1
660     PRINT "Set of parameters ";I
670     FOR J=1 TO G
680       PRINT J;
690       INPUT X[J,I]
700     NEXT J
710   NEXT I
720 DOEND
730 IF Q1=3 THEN 760
740 GOSUB 1520
750 END
760 INPUT "Initial simplexes to file? Y/N ",Q$
770 IF Q$="N" THEN 850
780 INPUT "Filename ",N2$
790 CREATE M,N2$,500
800 IF M=0 THEN 820
810 GOTO 780
820 ASSIGN N2$,2,M
830 IF M=0 OR M=1 THEN 850
840 PRINT "No output to file"
850 REM INPUT "Number of initial simplexes ",R9
860 R9=10
870 REDIM S9[ ,G+2]
880 MAT S9=ZER
890 FOR I=1 TO G
900   PRINT "Parameter ",I
910   INPUT "Minimum ",R1[I]
920   INPUT "Maximum ",R2[I]
930   R3[I]=R2[I]-R1[I]
940   INPUT "Initial simplex size as fraction of range ",R4[I]
950 NEXT I
960 FOR R=1 TO R9
970   PRINT "Initial simplex ";R;" **********"
980   C=0
990   C=C+1
1000  PRINT "Attempt ";C
1010  FOR I5=1 TO G
1020    X[I5,1]=(R3[I5]*RND(0)+R1[I5]-R4[I5])/(I5+1)
1030  NEXT I5
1040  FOR I5=1 TO G
1050    FOR J5=1 TO I5
1060      X[I5,J5]=X[I5,1]
1070    NEXT J5
1080    X[I5,I5+1]=X[I5,1]+R3[I5]*R4[I5]
1090  NEXT I5
1100  FOR I5=1 TO G-1
1110    X[I5,I5+2]=0
1120    FOR J5=1 TO I5+1
1130      X[I5,I5+2]=X[I5,I5+2]+X[I5,J5]
1140    NEXT J5
```

```
1150      X[I5,I5+2]=X[I5,I5+2]/(I5+1)
1160      FOR K=I5+3 TO G+1
1170        X[I5,K]=X[I5,I5+2]
1180      NEXT K
1190    NEXT I5
1200    FOR I5=1 TO G
1210      FOR J5=1 TO G+1
1220        IF X[I5,J5]<R1[I5] OR X[I5,J5]>R2[I5] THEN 990
1230      NEXT J5
1240    NEXT I5
1250    MAT PRINT X
1260    MAT X0=X
1270    GOSUB 1520
1280    IF Q$="Y" THEN DO
1290      IF C0=1 THEN M$="N"
1300      IF C0=0 THEN M$="C"
1310      GOSUB 4870
1320    DOEND
1330    IF C0=1 THEN DO
1340      C0=0
1350      C4=C4+1
1360      PRINT "Restart"
1370      C2=C2+1
1380      IF C2>5 THEN 1400
1390      GOTO 1010
1400      C2=0
1410    DOEND
1420  NEXT R
1430  PRINT
1440  IF Q1=1 OR Q1=2 THEN 1490
1450  PRINT "******* SUMMARY *******"
1460  PRINT "Parameters, Minimum, No of evaluations"
1470  MAT PRINT S9
1480  PRINT "Not converged ";C4
1490  PRINT "************************"
1500  END
1510  REM ========================== SIMPLEX ================
1520  I9=0
1530  REM Calculate center of gravity (used instead of centroid)
1540  MAT G2=ZER
1550  FOR I=1 TO G
1560    FOR J=1 TO G1
1570      G2[I]=G2[I]+X[I,J]
1580    NEXT J
1590  NEXT I
1600  MAT G2=(1/G1)*G2
1610  FOR I=1 TO G1
1620    FOR J=1 TO G
1630      T[J]=X[J,I]
1640    NEXT J
1650    Z=FNV(P,S,G,T[*],V1[*],X1[*],V[*],X[*,*])
1660    V[I]=S
1670  NEXT I
1680  Z=FNS(C5,G1,W,N,B,V[*])
1690  I9=I9+1
1700  REM F=1 is for reflection
1710  F=1
```

```
1720 Z=FNF(G,G1,W,F,G2[*],T[*],X[*,*])
1730 MAT T1=T
1740 Z=FNV(P,S,G,T[*],V1[*],X1[*],V[*],X[*,*])
1750 S1=S
1760 IF I9>I0 THEN GOTO 4920
1770 GOTO C5 OF 1780,1910
1780 REM if better than best look at expansion
1790 IF S<V[B] THEN 2230
1800 REM if better than next-to-worst,substitute for worst
1810 GOTO C3 OF 1820,1840
1820 IF S<V[N] THEN 2330
1830 GOTO 1850
1840 IF S<V[W] THEN 2330
1850 REM if better than worst look at positive contraction
1860 GOTO C3 OF 1870,1890
1870 IF S<V[W] THEN 2430
1880 GOTO 2030
1890 IF S<V[N] THEN 2430
1900 GOTO 2030
1910 REM if better than best look at expansion
1920 IF S>V[B] THEN 2230
1930 REM if better than next-to-worst,substitute for worst
1940 GOTO C3 OF 1950,1970
1950 IF S>V[N] THEN 2330
1960 GOTO 1990
1970 IF S>V[W] THEN 2330
1980 REM if better than worst look at positive contraction
1990 GOTO C3 OF 2000,2020
2000 IF S>V[W] THEN 2430
2010 GOTO 2030
2020 IF S>V[N] THEN 2430
2030 REM F=-.5 for negative contraction
2040 F=-.5
2050 Z=FNF(G,G1,W,F,G2[*],T[*],X[*,*])
2060 Z=FNV(P,S,G,T[*],V1[*],X1[*],V[*],X[*,*])
2070 REM if no better than worst reflect from N
2080 GOTO C5 OF 2090,2140
2090 GOTO C3 OF 2100,2120
2100 IF S>V[W] THEN 2670
2110 GOTO 2180
2120 IF S>V[N] THEN 2670
2130 GOTO 2180
2140 GOTO C3 OF 2150,2170
2150 IF S<V[W] THEN 2670
2160 GOTO 2180
2170 IF S<V[N] THEN 2670
2180 Z=FNR(G,G1,W,B,S,G2[*],T[*],V[*],X[*,*])
2190 PRINT "NC ";
2200 REM reflect from next-to-worst
2210 GOTO 2790
2220 REM ------------------------------------------
2230 F=2
2240 Z=FNF(G,G1,W,F,G2[*],T[*],X[*,*])
2250 Z=FNV(P,S,G,T[*],V1[*],X1[*],V[*],X[*,*])
2260 REM if expansion better than reflection, make it
2270 GOTO C5 OF 2280,2300
2280 IF S<S1 THEN 2380
```

```
2290 GOTO 2310
2300 IF S>S1 THEN 2380
2310 S=S1
2320 REM make reflection and go to convergence test
2330 Z=FNR(G,G1,W,B,S,G2[*],T1[*],V[*],X[*,*])
2340 PRINT "R ";
2350 GOTO 2790
2360 REM ------------------------------------------
2370 REM make expansion and go to convergence test
2380 Z=FNR(G,G1,W,B,S,G2[*],T[*],V[*],X[*,*])
2390 PRINT "E ";
2400 GOTO 2790
2410 REM ------------------------------------------
2420 REM F=.5 for positive contraction
2430 F=.5
2440 Z=FNF(G,G1,W,F,G2[*],T[*],X[*,*])
2450 Z=FNV(P,S,G,T[*],V1[*],X1[*],V[*],X[*,*])
2460 REM if better than next-to-worst, make it
2470 GOTO C5 OF 2480,2530
2480 GOTO C3 OF 2490,2510
2490 IF S<V[N] THEN 2620
2500 GOTO 2570
2510 IF S<V[W] THEN 2620
2520 GOTO 2570
2530 GOTO C3 OF 2540,2560
2540 IF S>V[N] THEN 2620
2550 GOTO 2570
2560 IF S>V[W] THEN 2620
2570 REM make it anyway and go to reflection from next-to-worst
2580 Z=FNR(G,G1,W,B,S,G2[*],T[*],V[*],X[*,*])
2590 PRINT "P";
2600 GOTO 2710
2610 REM ------------------------------------------
2620 Z=FNR(G,G1,W,B,S,G2[*],T[*],V[*],X[*,*])
2630 PRINT "PC ";
2640 GOTO 2790
2650 REM ------------------------------------------
2660 REM Going to reflect from N but first replacing W regardless
2670 Z=FNR(G,G1,W,B,S,G2[*],T[*],V[*],X[*,*])
2680 PRINT "N";
2690 REM ------------------------------------------
2700 REM reflection from next-to-worst
2710 N1=N
2720 N=W
2730 W=N1
2740 PRINT "rn ";
2750 C3=2
2760 GOTO 1690
2770 REM ------------------------------------------
2780 REM convergence check
2790 Z=FNT(C5,G1,V[*],D[*],D1[*])
2800 FOR I=1 TO 4
2810   IF ABS(D1[I])>.00001 THEN 2880
2820   GOTO C5 OF 2830,2850
2830   IF D1[I]>0 THEN 4920
2840   GOTO 2860
2850   IF D1[I]<0 THEN 4920
```

```
2860 NEXT I
2870 GOTO 2990
2880 C3=1
2890 Z=FND(G,G1,V[*],D)
2900 IF I9>I0 THEN GOTO 4920
2910 IF I9 MOD I7=0 THEN DO
2920   PRINT
2930   PRINT "# ";I9
2940   GOSUB 4810
2950 DOEND
2960 IF D>C1 THEN 1680
2970 Z=FNG(G,G1,V[*],X[*,*],G2[*],D)
2980 IF D>C1 THEN 1680
2990 Z=FNS(C5,G1,W,N,B,V[*])
3000 PRINT
3010 PRINT " ***********> CONVERGED <***************"
3020 PRINT "After ";I9;" function evaluations"
3030 PRINT "Optimized parameters"
3040 PRINT "Best"
3050 H=B
3060 GOSUB 4600
3070 PRINT "Worst"
3080 H=W
3090 GOSUB 4600
3100 PRINT "------------"
3110 GOSUB 4810
3120 IF Q1=1 OR Q1=2 THEN 3180
3130 FOR I=1 TO G
3140   S9[R,I]=X[1,B]
3150 NEXT I
3160 S9[R,G+1]=V[B]
3170 S9[R,G+2]=I9
3180 RETURN
3190 REM ============== Functions definitions ============
3200 DEF FND(G,G1,V[*],LONG D)
3210   REM Standard deviation of centroid vertices in conv.test
3220   INTEGER I
3230   REAL S1,S2
3240   S1=S2=0
3250   FOR I=1 TO G1
3260     S1=S1+V[I]
3270     S2=S2+V[I]*V[I]
3280   NEXT I
3290   D=SQR(ABS(((G1)*S2-S1*S1)/(G*G1)))
3300   RETURN D
3310 FNEND
3320 DEF FNF(G,G1,W,F,G2[*],T[*],X[*,*])
3330   REM Find coordinates, type depending on F
3340   INTEGER I
3350   FOR I=1 TO G
3360     T[I]=G1*(1+F)*G2[I]/G-(1+G1*F)*X[I,W]/G
3370   NEXT I
3380   RETURN T[1]
3390 FNEND
3400 DEF FNG(G,G1,V[*],X[*,*],G2[*],LONG D)
3410   REM Find response in the center of gravity & its deviation
3420   REM from vertices
```

```
3430   INTEGER I
3440   REAL S
3450   Z=FNV(P,S,G,G2[*],V1[*],X1[*],V[*],X[*,*])
3460   D=0
3470   FOR I=1 TO G1
3480     D=D+(V[I]-S)*(V[I]-S)
3490   NEXT I
3500   D=SQR(ABS(D/G1))
3510   RETURN D
3520 FNEND
3530 DEF FNR(G,G1,W,B,S,G2[*],T[*],V[*],X[*,*])
3540   REM Replace worst vertex by coordinates in T[*]
3550   REM and update center of gravity
3560   INTEGER I
3570   FOR I=1 TO G
3580     G2[I]=G2[I]+(T[I]-X[I,W])/G1
3590     X[I,W]=T[I]
3600   NEXT I
3610   V[W]=S
3620   B=W
3630   RETURN X[1,1]
3640 FNEND
3650 DEF FNS(C5,G1,W,N,B,V[*])
3660   REM Sort responses and find Worst, Next-to-worst, and Best
3670   INTEGER I
3680   REAL M1,M2
3690   M1=-1E70
3700   M2=1E70
3710   FOR I=1 TO G1
3720     IF V[I]<M1 THEN 3750
3730     M1=V[I]
3740     W=I
3750     IF V[I]>M2 THEN 3780
3760     M2=V[I]
3770     B=I
3780   NEXT I
3790   IF C5=2 THEN DO
3800     N=W
3810     W=B
3820     B=N
3830   DOEND
3840   GOTO C5 OF 3850,3930
3850   M1=-1E70
3860   FOR I=1 TO G1
3870     IF I=W THEN 3910
3880     IF V[I]<M1 THEN 3910
3890     M1=V[I]
3900     N=I
3910   NEXT I
3920   GOTO 4000
3930   M2=1E70
3940   FOR I=1 TO G1
3950     IF I=W THEN 3990
3960     IF V[I]>M2 THEN 3990
3970     M2=V[I]
3980     N=I
3990   NEXT I
```

```
4000   RETURN W
4010 FNEND
4020 DEF FNT(C5,G1,V[*],D[*],D1[*])
4030   REM Mean response and its difference "stack"
4040   INTEGER I
4050   D[1]=D[2]
4060   D[2]=D[3]
4070   D[3]=D[4]
4080   D[4]=D[5]
4090   D[5]=0
4100   FOR I=1 TO G1
4110     D[5]=D[5]+V[I]
4120   NEXT I
4130   D[5]=D[5]/G1
4140   D1[1]=D[2]-D[1]
4150   D1[2]=D[3]-D[2]
4160   D1[3]=D[4]-D[3]
4170   D1[4]=D[5]-D[4]
4180   RETURN D[1]
4190 FNEND
4200 DEF FNV(P,S,G,T[*],V1[*],X1[*],V[*],X[*,*])
4210   REM Calculate function value
4220   INTEGER I
4230   S=0
4240   REM ************************************************************
4250   REM                    EVALUATED FUNCTION COMES IN HERE      *
4260   REM Must return value in S
4270   REM FOR I=1 TO P
4280   REM S=S+(V1[I]-T[1]*(1-EXP(-T[2]*X1[I])))**2
4290   REM NEXT I
4300   REM S=T[1]*T[1]+T[2]*T[2]
4310   REM S=100*(T[2]-T[1])*(T[2]-T[1])+(1-T[1])*(1-T[1])
4320   REM IF T[1]>0 THEN Q=ATN(T[2]/T[1])/6.28318
4330   REM IF T[1]<0 THEN Q=(6.28318+ATN(T[2]/T[1]))/6.28318
4340   REM S=100*(T[1]-10*Q)**2+((T[1]**2+T[2]**2)**.5 -1)**2
4350   REM S=S+T[2]**2
4360   S=2*T[1]**3-3*T[1]**2-2*T[2]**3+3*T[2]**2
4370   REM S=2-(T[1]-1)**2-(T[2]-3)**2
4380   REM ************************************************************
4390   RETURN S
4400 FNEND
4410 REM ================I/O Subroutines ========================
4420 IF Q9=0 THEN 4450
4430 PRINT "File problem "
4440 STOP
4450 LINPUT #1;B$
4460 B1$=B$[1;3]
4470 IF B1$="PAR" THEN 4660
4480 REM Initial parameter file
4490 B1$=DEB$(B1$)
4500 CONVERT B1$ TO P
4510 FOR I=1 TO P
4520   LINPUT #1;B$
4530   B1$=B$[1;14]
4540   CONVERT B1$ TO X1[I]
4550   B1$=B$[15;14]
4560   CONVERT B1$ TO V1[I]
```

```
4570 NEXT I
4580 RETURN
4590 REM Output subroutine
4600 FOR I=1 TO G
4610   PRINT X[I,H];
4620 NEXT I
4630 PRINT
4640 PRINT "Optimized value ";V[H]
4650 RETURN
4660 LINPUT #1;B$
4670 B1$=B$[1;2]
4680 B1$=DEB$(B1$)
4690 CONVERT B1$ TO G
4700 G1=G+1
4710 FOR I=1 TO G
4720   LINPUT #1;B$
4730   FOR J=1 TO G+1
4740     J1=5*J-4
4750     IF J1>65 THEN J1=J1-64
4760     B1$=B$[J1;5]
4770     CONVERT B1$ TO X[I,J]
4780   NEXT J
4790 NEXT I
4800 RETURN
4810 PRINT
4820 MAT PRINT X
4830 PRINT "--------"
4840 MAT PRINT V
4850 RETURN
4860 REM ---------------------------------------
4870 PRINT #2;M$
4880 MAT PRINT #2;X0
4890 PRINT #2;V[B]
4900 RETURN
4910 REM -----------------------------------
4920 PRINT
4930 PRINT "Not converged after ";I9;" function evaluations"
4940 GOSUB 4810
4950 C0=1
4960 GOTO 3180
SAVE SX!
EXIT
```

Appendix 2 — Example of interactive input of parameters and output of intermediate results.

```
:RUN SXC

Desired function in FNV? (Y/N). IF NOT, c-Y or BRK ]
Data from file (1), manually (2), not needed (3) 3
Initial simplex from file(1),manually (2),computer(3) 3
Number of parameters 2
Initial simplexes to file? Y/N N
Parameter       1
Minimum -100
Maximum 100
Initial simplex size as fraction of range .05
Parameter       2
Minimum -100
Maximum 100
Initial simplex size as fraction of range .05
Initial simplex  1     *********
Attempt  1
 42.5637         52.5637         47.5637

 27.2918         27.2918         37.2918

E E R PC NC Prn Prn NC E NC R E R E NC R NC NC NC Nrn R E NC NC NC R N
rn NC Nrn NC R NC R NC NC E R E NC Nrn R R NC R NC R NC R NC E R R R NC
R Nrn NC NC NC R
#  60

 .692031         .6953           .683329

 .321541         .323852         .312119

--------
 .159523         .159401         .1595

NC NC
 ***********> CONVERGED <**************
After  62    function evaluations
Optimized parameters
Best
 .688164      .316951
Minimized value  .159345
Worst
 .6953        .323852
Minimized value  .159401
------------
                  .               .               .
                  .               .               .
                  .               .               .
                  .               .               .

****** SUMMARY ******
Parameters, Minimum, No of evaluations
 .688164         .316951         .159345         62
```

.688512	.317428	.159341	48
.687781	.316732	.159354	56
.694594	.323637	.159401	59
.691574	.32074	.159379	59
.69364	.322438	.15935	51
.694736	.323439	.159374	52
.690261	.318711	.159369	52
.686567	.315556	.159381	61
.69359	.322634	.159379	53

Not converged 0
****

END OF PROGRAM

POOR

QUALITY

ORIGINAL